

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-244993

(P2001-244993A)

(43) 公開日 平成13年9月7日 (2001.9.7)

(51) Int.Cl. ⁷	識別記号	F I	デマコト* (参考)
H 0 4 L 12/56	3 0 0	H 0 4 L 12/56	3 0 0 A
	2 3 0		2 3 0 Z
12/28		11/00	3 1 0 B
29/06		13/00	3 0 5 A

審査請求 未請求 請求項の数4 O L (全 14 頁)

(21) 出願番号 特願2001-14224(P2001-14224)

(22) 出願日 平成13年1月23日 (2001.1.23)

(31) 優先権主張番号 09/497002

(32) 優先日 平成12年2月2日 (2000.2.2)

(33) 優先権主張国 米国 (US)

(71) 出願人 59607/259

ルーセント テクノロジーズ インコーポ
レイテッドLucent Technologies
Inc.アメリカ合衆国 07974 ニュージャージ
ー、マレーヒル、マウンテン アベニュー
600-700

(72) 発明者 ムーイ チュー チュア

アメリカ合衆国、07746 ニュージャージ
ー、マルボロ、スカイラーシ コート 1

(74) 代理人 100081053

弁理士 三俣 弘文

最終頁に続く

(54) 【発明の名称】 パケットサーバで用いられる通信方法

(57) 【要約】

【課題】 UMTSコアネットワークにおいて、GTP／UDP／IPヘッダの圧縮により、小さいマルチメディアRTPパケットをより効率的にトランスポートする。

【解決手段】 GPRS (General Packet Radio Service) に基づくトンネリングプロトコル (GTP)、ユーザデータグラムプロトコル (UDP)、およびインターネットプロトコル (IP) のヘッダ情報を含む GTP／UDP／IPヘッダの圧縮を GTP ピアと ネゴシエートし、まず GTP ピアへ GTP／UDP／IPヘッダを送信し、次に圧縮 GTP／UDP／IPヘッダを GTP ピアへ送信する。圧縮 GTP／UDP／IPヘッダは、長さ 4 バイト以下のトンネル識別子フィールド、あるいは、拡張フィールドの有無を示す拡張ビットフィールドと、トンネル識別子フィールドの有無を示すトンネル識別子存在フィールドとを有する。

圧縮 GTP ヘッダ

Vers	PT	E	S	T	N
Msg TYPE (1バイト)					
L LENGTH (2バイト)					
TID (2または4バイト)					
シーケンス番号(2バイト)					
拡張タイプ		拡張ヘッダ長			
拡張内容					

拡張ヘッダ

Vers: バージョン
P: ペイロードタイプ
E: 拡張ビット
S: シーケンス番号
T: TID 存在
N: SNDCP 存在
Msg: メッセージ
TID: トンネル識別子

【特許請求の範囲】

【請求項1】 GPRS (General Packet Radio Service) に基づくトンネリングプロトコル (GTP)、ユーザデータグラムプロトコル (UDP)、およびインターネットプロトコル (IP) のヘッダ情報を含む GTP/UDP/IP ヘッダの圧縮を GTP ピアとネゴシエートするステップと、

まず GTP ピアへ GTP/UDP/IP ヘッダを送信するステップと、

次に圧縮 GTP/UDP/IP ヘッダを GTP ピアへ送信するステップとを有することを特徴とする、パケットサーバで用いられる通信方法。

【請求項2】 圧縮 GTP/UDP/IP ヘッダは、長さ4バイト以下のトンネル識別子フィールドを少なくとも有することを特徴とする請求項1記載の方法。

【請求項3】 圧縮 GTP/UDP/IP ヘッダは、少なくとも、拡張フィールドの有無を示す拡張ビットフィールドと、トンネル識別子フィールドの有無を示すトンネル識別子存在フィールドとを有することを特徴とする請求項1記載の方法。

【請求項4】 UDP および IP のヘッダ情報を有する RTP (Real Time Protocol) に従ってデータパケットをフォーマットするステップと、

パケットを送信する前に RTP/UDP/IP ヘッダを圧縮するステップとをさらに有し、

圧縮 RTP/UDP/IP ヘッダの1つのフィールドは、UDP チェックサムフィールドの有無を定義することを特徴とする請求項1記載の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、通信に関し、特に、パケット通信システムに関する。

【0002】

【従来の技術】 ワイヤレスシステムが発展するとともに、移動通信交換センタ (MSC) と基地局の間の通信は、インターネットプロトコル (IP) に基づくトランスポートメカニズムに移行しつつある。(ここで、ワイヤレスシステムという用語は、例えば、CDMA (符号分割多元接続)、GSM (Global System for Mobile Communications)、提案されている UMTS (Universal Mobile Telecommunications System) などを指す。) ワイヤレス通信の性質 (例えば、リアルタイム音声) を考慮すると、IP ベースのトランスポートは、リアルタイムアプリケーションを収容するプロトコルを利用する必要がある。

【0003】 そのようなプロトコルの1つは、RTP (Real Time Protocol) である (例えば、H. Schulzrinne, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, を参照)。RTP が注目されているのは、それが、リアルタ

イムストリームを扱うために利用可能な IETF (Internet Engineering Task Force) プロトコルであるためである。RTP トラフィックは、UDP (ユーザデータグラムプロトコル)、および IP パケットにカプセル化される。

【0004】

【発明が解決しようとする課題】 残念ながら、RTP/UDP/IP を使用すると、voice-over-IP アプリケーションがワイヤレスネットワークで実行される際に大きいオーバーヘッドが生じる。音声ペイロードは通常小さい (例えば、10 から 20 バイト) が、RTP/UDP/IP ヘッダは 40 バイトであるためである。

【0005】

【課題を解決するための手段】 RTP/UDP/IP ヘッダに伴う大きいオーバーヘッド以外にも、この状況は、GTP (General Packet Radio Service Tunneling Protocol) カプセル化パケットを使用することによってさらに悪化する。この場合、GTP/UDP/IP オーバーヘッドは、10 バイトの音声ペイロードの約 980 % となる。そこで、本発明によれば、GTP/UDP/IP ヘッダは送信のために圧縮される。

【0006】 本発明の実施例では、UMTS コアネットワークは、GTP/UDP/IP ヘッダの圧縮 (以下、「GTP ヘッダ圧縮」あるいは「圧縮 GTP ヘッダ」という) を行う圧縮フレームワークをサポートする。さらに、UMTS コアネットワークは、GTP ヘッダ圧縮とは独立に、RTP/UDP/IP ヘッダ圧縮 (以下、「RTP ヘッダ圧縮」あるいは「圧縮 RTP ヘッダ」という) もサポートする。その結果、UMTS コアネットワークは、小さいマルチメディア RTP パケットをより効率的にトランスポートすることが可能となる。

【0007】

【発明の実施の形態】 当業者に知られている (リリース 97 バージョン) 非圧縮 (圧縮されていない) GTP カプセル化 RTP パケット 10 の例示的フォーマットを図 1 に示す。GTP/UDP/IP ヘッダ 11 は、IP/UDP ヘッダ 12 および GTP ヘッダ 13 からなる。GTP/UDP/IP ヘッダ 11 は、当業者に知られているように、GTP ペイロード 14 の最上位に位置する。例えば 10 バイトに等しい音声ペイロード (例えば、図 1 のペイロード 15) の場合、GTP/UDP/IP ヘッダ 11 の結果として、オーバーヘッドは約 980 % となる。そこで、本発明によれば、GTP/UDP/IP ヘッダは送信のために圧縮される。図 1 からわかるように、GTP ペイロード 14 は、IP ヘッダおよび UDP ヘッダもトランスポートする。当業者に知られているように、IP/UDP ヘッダ 12 は、トンネルに関する発信側および着信側情報を含むが、GTP ペイロード 14 の IP ヘッダおよび UDP ヘッダは、通信エンドポイントに関する発信側および着信側情報を含む。(注意すべ

き点であるが、TID値は、"Global System for Mobile Communications (GSM) 04.08 document, Digital cellular telecommunications system (Phase 2+); mobile radio interface layer 3 - specification"で定義されたリリース97バージョンに基づいている。TID値は、以下の8バイトを含む：MCC 12ビット、MNC 8ビット、MSIN 40ビットおよびNSAPI 4ビット。MCC、MNC、MSINおよびNSAPIは、GSM 04.08 documentで定義されたIMSIの一部である。）

【0008】本発明の原理に従って修正された例示的なUMTSネットワーク200を図2に示す。本発明の概念以外は、図2に示した要素は周知であり、詳細に説明しない。例えば、UMTSネットワーク200は、無線アクセスネットワーク（RAN：radio access network）、コアネットワーク（CN：core network）、バックボーンネットワーク、および、IPエンドホスト240によって表される例示的なエンドポイントを有する。バックボーンネットワークは、インターネットおよび公衆交換電話網（PSTN）を含む。RANは、移動局（MS）205、ノードB210および無線ネットワークコントローラ（RNC：radio network controller）215を有する。（UMTSは「ノードB」（node B）という用語を使用しているが、これは基地局ともいう。）CNは、サービスGPRSサポートノード（SGSN：serving GPRS support node）220、ゲートウェイGPRSサポートノード（GGSN：gateway GPRS support node）225および要素230を有する。要素230は、ゲートキーパ（GK）（ITU H. 323のコンポーネント）およびIP/PSTNゲートウェイ（GW）（H. 323とPSTNの間の翻訳のため）を有する。図では、UMTSネットワーク200の要素は、単一ブロックの要素として示されているが、蓄積プログラム制御プロセッサ、メモリ、および適当なインタフェースカード（図示せず）を有する。本発明の説明のため、あるエンドツーエンドコネクションがMS205とIPエンドホスト240（これはエンドポイントともいう）の間にあるとする。以下、「パケットサーバ」という用語は、任意のパケットプロセッサを指す。その例は、上記のUMTS200の要素である。

【0009】本発明によれば、UMTSネットワーク200は、GTPヘッダ圧縮を行う圧縮フレームワークをサポートする。さらに、UMTSネットワーク200は、GTPヘッダ圧縮とは独立に、RTPヘッダ圧縮もサポートする。（ここで、「GTPヘッダ圧縮」あるいは「圧縮GTPヘッダ」という用語は、GTP/UDP/IPヘッダの圧縮を指す。同様に、「RTPヘッダ圧縮」あるいは「圧縮RTPヘッダ」という用語は、RTP/UDP/IPヘッダの圧縮を指す。）GTPヘッダ圧縮は、RTPヘッダ圧縮とは独立であるため、GTP

ピアは、RTPピアとは異なることが可能である（しかしこれが要求されるわけではない）。これは、多少の設計の自由度（フレキシビリティ）も提供する。マルチメディアトラフィックには、RTPを使用せず、純粋にUDPカプセル化を使用するものもあるからである。その結果、UMTSネットワーク200は、小さいマルチメディアパケットをより効率的にトランスポートすることができる。以下で、本発明の概念について説明する。

【0010】ヘッダ圧縮の概観

理解されるべき点であるが、本発明によれば、RTPヘッダ圧縮およびGTPヘッダ圧縮は、ピア間で独立にネゴシエートすることができる（詳細は後述）。本発明の概念以外については、圧縮/伸長技術は周知であり、ここでは説明しない。例えば、通常、圧縮/伸長を行うもの（コンプレッサ(compressor)/デコンプレッサ(decompressor)）は、例えばMS205のメモリ（図示せず）に記憶されたソフトウェアモジュールであり、例えばMS205の蓄積プログラム制御マイクロプロセッサ（図示せず）によって実行される。このソフトウェアモジュールは、従来のプログラミング技法（したがって、同じくここでは説明しない）を用いて、共有情報を記憶し（後述）、伝送用に圧縮（短縮）された形式のGTP/UDP/IPヘッダあるいはRTP/UDP/IPヘッダを形成（フォーマット）する。

【0011】移動局（例えば、図2のMS205）に関して、コンプレッサ/デコンプレッサを有する2つの例示的なプロトコルスタックを図3および図4に示す。図3のプロトコルスタックでは、RTPコンプレッサ/デコンプレッサは、IP層とRLC/MAC（無線リンク制御/メディアアクセス制御）リンク層との間に位置する。（簡単のため、プロトコルスタックの物理層は図示していない。）図4のプロトコルスタックでは、RTPコンプレッサ/デコンプレッサは、アプリケーション層とRLC/MACリンク層との間に位置する。（これも、物理層は図示していない。）（RTPヘッダ圧縮の形式については以下で説明するが、注意すべき点として、図4の実施例は、RTPヘッダが単に完全に除去されるようなRTPヘッダ圧縮の形式も表している（RTPヘッダは図13に例示する）。）

【0012】相補的に、対応するRTPコンプレッサ/デコンプレッサおよびGTPコンプレッサ/デコンプレッサが、UMTSネットワーク内に配置される。UMTSネットワーク200におけるRTPコンプレッサ/デコンプレッサおよびGTPコンプレッサ/デコンプレッサの配置例を図5に示す。図5において、RTPコンプレッサ/デコンプレッサ（C/D）は、MS205およびIPエンドホスト240内に配置され（すなわち、MS205とIPエンドホスト240はRTPピアである）、GTPコンプレッサ/デコンプレッサ（C/D）は、RNC215およびGGSN225内に配置される

(すなわち、RNC 215とGGSN 225はGTPピアである)。

【0013】RTPは、ポイントツーポイントプロトコルである。したがって、RTPヘッダ圧縮ピアにとって注意すべき点であるが、このことは、リンク層識別子(ID)が各RTPセッション識別子にマップされると仮定している。本発明の実施例のRTPセッション識別子は、IP終点(宛先、デスティネーション)アドレスおよびIP終点ポート番号(これらはエンドポイントのものである)、SSRC識別子(例えば、図13を参照)、ならびに、UDP終点ポート(例えば、図12を参照)を含む。ATM(非同期転送モード)がトランスポートとして使用される場合、例示的なリンク層IDは、対応するVPI/VC(仮想パス識別子/仮想コネクション識別子)である。リンク層IDとRTPセッションのマッピングは、各RTPピア内で行われる。

【0014】また、注意すべき点であるが、RTPコンプレッサ/デコンプレッサは、別法として、無線アクセスネットワーク内に(例えば、RNC 215内に)、あるいは、コアネットワーク内に(例えば、SGSN 220)に置くことも可能である。

【0015】コアネットワーク内では、GGSN 225にGTPコンプレッサ/デコンプレッサを置くことが(要求されるわけではないが)好ましい。GTPコンプレッサ/デコンプレッサがSGSN 220に配置される場合、SRNS(サービス無線ネットワークサブシステム: Serving Radio Network Subsystem)再配置によるハンドオーバー(「ハンドオフ」ともいう)も考慮しなければならないことがあるからである。(UMTSで知られているように、SRNSは、特定のRANだけでなく、サポート要素(例えばデータベース(図示せず))をも含む。)しかし、GTPコンプレッサ/デコンプレッサがGGSN 225に配置される場合、SRNS再配置の場合でもコンテキスト転送は不要である。

【0016】次に、図6~図10に、UMTSネットワーク200でGTPヘッダ圧縮およびRTPヘッダ圧縮を使用するための例示的なメッセージフローを示す。本発明の概念以外は、以下の説明では周知のUMTSメッセージフローを利用し、これについては説明しない。図6~図10では、GTPヘッダピアはRNC 215とGGSN 225であり、RTPヘッダピアはMS 205とIPエンドホスト240であると仮定する。

【0017】図6に、図2および図5の移動局(例えば、MS 205)が、どのようにしてGTPヘッダ圧縮/伸長をネゴシエートするかを示す。"Attach Procedures"がMS 205とRNC 215の間で実行(当業者に知られているように)された後、MS 205は、本発明に従って、あらかじめ定義された識別子"GTP_Comp"によって表されるGTPヘッダ圧縮要求を含むように修正された"Activate PDP context" request (PDP(packet

data protocol)コンテキスト起動要求)メッセージをSGSN 220へ送信する。これに応答して、SGSN 220は、GTPヘッダ圧縮に対する要求を意味する"Create PDP context" request (PDPコンテキスト作成要求)メッセージ("GTP_Comp"識別子を含むように修正したもの)をGGSN 225へ送る。GGSN 225は、確認応答(acknowledgment)として、"Create PDP context" response (PDPコンテキスト作成応答)メッセージ("GTP_Comp"識別子を含むように修正したもの)で応答する。SGSN 220は、この"Create PDP context" responseメッセージを受信すると、"Activate PDP context" response (PDPコンテキスト起動応答)メッセージ("GTP_Comp"識別子を含むように修正したもの)をMS 205に送る。

【0018】上記のように、GTPヘッダ圧縮コンテキストを設定するためには、GTP_Compressed (GTP圧縮)フラグ(例えば、あらかじめ定義されたビットパターン)またはGTP Header Compression_Context IE (GTPヘッダ圧縮コンテキスト情報エレメント (IE: information element))を、従来のメッセージセットに追加する。このGTP Header Compression_Context IEは、GTPの完全なヘッダ情報を含むことになる。このエレメントが存在する場合、GTPヘッダコンテキストを設定するために完全なGTPヘッダを送る必要はない。そうでない場合、GTPヘッダコンテキストを設定するために、完全なGTPヘッダとともに1個以上のパケットを送信する必要がある場合がある。(注意すべき点であるが、GTPコンプレッサ/デコンプレッサがSGSNに配置され、SRNS再配置の結果SGSNの変更が生じる場合、新SGSNは旧SGSNにGTPコンテキスト問合せメッセージを送り、旧SGSNが適当なGTPコンテキスト応答メッセージで応答することにより、新SGSNがGTPヘッダ圧縮のための新たなコンプレッサ/デコンプレッサポイントとなることができる。

【0019】次に、図7に、例示的なパケットフローを示す。GTPヘッダ圧縮の場合、GTPピア(ここでは、RNC 215とGGSN 225によって表される)間でGTP圧縮ヘッダコンテキストを確立する(図7、(A))ために、完全なGTPヘッダを有する少なくとも1つのパケットを送信する。前述のように、パケットは、GGSN 225とRNC 215の間でGTPを用いて通信される(すなわち、GTPは、GGSN 225およびRNC 215で終端する)。GTPヘッダ圧縮がネゴシエートされた後、各GTPピア(例えば、GGSN 225)は、GTPに従ってパケットをフォーマットしてから、そのGTPピア(ここではRNC 215)へパケットを送信する前にGTPヘッダを圧縮し(後述)、このGTPピアは、GTPヘッダを圧縮解除して、MS 205へ送信するためにペイロード(これは、RTPを

含むことも含まないこともある)を回復する。同様に、逆向きには、RNC215は、GGSN225へ送信するためにGTPヘッダを圧縮し、GGSN225は、GTPヘッダを圧縮解除して、ペイロードを回復する。パケットは、当業者に知られているように、リンク層で、GGSN225とIPエンドホスト240の間で通信される。GTPヘッダ圧縮ネゴシエーションに続き、オプションとして、図6のGTPヘッダ圧縮ネゴシエーションの場合に示したのと同様に、RTPヘッダ圧縮を、MS205とIPエンドホスト240の間でネゴシエートすることも可能である(図7、(B))。

【0020】RTPヘッダ圧縮コンテキストメッセージ交換の例を図8に示す。(この場合も、従来のシグナリングメッセージの修正を仮定する。したがって、追加のメッセージ要件、例えば、これが“RTP context set up”(RTPコンテキスト設定)メッセージであることを識別するために、あらかじめ定義されたビット値が追加される。)最初に、2つのRTPピアは、RTPヘッダ圧縮コンテキストを設定するためにシグナリングメッセージを交換する。“RTP context set up” request (RTPコンテキスト設定要求)メッセージが、一方のRTPピア(例えば、MS205)から、他方のRTPピア(例えば、IPエンドホスト240)に送られる。“RTP context set up” response (RTPコンテキスト設定応答)メッセージが、ハンドシェイクを完了する。RTPコンテキストに変更がある場合、RTP圧縮ヘッダ内で運ばれる追加変更(差分、デルタ)情報を示すために、適当なコンテキスト更新コードが圧縮RTPヘッダ(後述)の第1バイトに用いられる。このように、“RTP context update” request (RTPコンテキスト更新要求)メッセージは、RTP圧縮ヘッダ内のインプリシット(暗黙的)なメッセージである。しかし、“RTP context update” response (RTPコンテキスト更新応答)メッセージは、オプションである。(注意すべき点であるが、2つのRTPピア(ここでは、MS205およびIPエンドホスト240によって表される)の間のRTPヘッダ圧縮は、帯域外シグナリングメッセージを交換してRTPヘッダ圧縮を開始することも可能である。この場合も、GTPヘッダ圧縮とRTPヘッダ圧縮は互いに独立であるため、必ずしもRTPヘッダ圧縮をネゴシエートする必要はない(その場合、パケットは、圧縮GTPヘッダおよび非圧縮RTPペイロードからなる)。同様に、RTPヘッダ圧縮を使用するかどうかにかかわらず、必ずしもGTPヘッダ圧縮をネゴシエートする必要はない。)

【0021】図7に戻り、RTPヘッダ圧縮コンテキストが設定された後、GTPヘッダ圧縮およびRTPヘッダ圧縮を用いて、後続のパケットが送信される(図7、(C)) (上記のように、GTPヘッダ圧縮およびRTPヘッダ圧縮が両方とも開始されると仮定する)。RT

Pコンテキスト再同期が要求される場合、受信側RTPピアは、“RTP context repair”(RTPコンテキスト修復)メッセージを送信側RTPピアへ送ることができる。これを、図9、(D)に示す。ここで、受信側RTPピアはIPエンドホスト240によって表され、送信側RTPピアは、MS205によって表される。その後、送信側RTPピアは、完全なヘッダを有する1個以上のRTPパケットを送信(図9(E))した後、圧縮RTPパケットを送信する(図9(F))。

【0022】同様に、GTPパケットが損失した場合、受信側GTPピアは、“GTP context repair”(GTPコンテキスト修復)メッセージを送信側GTPピアへ送ることができる。これを図10、(G)に示す。ここで、受信側GTPピアはGGSN225によって表され、送信側GTPピアはRNC215によって表される。その後、送信側GTPピアは、完全なGTPヘッダを有する1個以上のパケットを送信(図10、(H))した後、圧縮GTPヘッダを有するパケットを送信する(図10、(I))。(理解されるように、この例では、RTPヘッダ圧縮同期は喪失していない。)

【0023】GTPヘッダ圧縮あるいはRTPヘッダ圧縮のいずれに対する上記のコンテキスト修復メカニズムも、損失パケットがあれば実行される。注意すべき点であるが、再同期のために明示的なコンテキスト修復メッセージが一定期間後に送信側へ送られるように、あらかじめ定義された期間しきい値を設定することも可能である。)

【0024】いずれかのGTPピアがGTPコンテキストを破棄したい場合、GTPコンテキスト破棄メッセージ(図示せず)を送信することができる。同様に、RTPピアは、RTPコンテキスト破棄メッセージ(図示せず)の送信により、RTPコンテキストを破棄することができる。

【0025】RTPヘッダ圧縮

UMTS環境に直接に適用可能ではないが、40バイトのRTP/UDP/IPヘッダを4~5バイトに短縮する従来のいくつかの提案がある(例えば、

・S. Casner and V. Jacobson, “Compressing IP/UDP/RTP Headers for Low-Speed Serial Links”, IETF RFC2508

・L. Jonsson, M. Degermark, H. Hannu and K. Svanbro, “Robust Checksum-based Header Compression”, IETF Internet draft, Oct 1999

を参照)。そこで、図11~図13に、従来のIP、RTPおよびUDPのヘッダフォーマットを参考のために示すが、これらについて詳細にはここでは説明しない。

【0026】一般に、IPヘッダに関しては(現在使用されているIPバージョンであるIPv4の使用を仮定する)、通常、全長(パケット長、total length)、パケットID(識別)およびヘッダチェックサムフィールド

ドのみが変化する。しかし、全長は、リンク層によっても提供されるため、冗長である。パケットIDは通常、各パケットごとに1または小さい数だけインクリメントされる。仮にIPパケット断片化がないとすると、このフィールドも通信する必要がなくなる。しかし、無損失圧縮を維持するために、パケットIDの変化を送信することも可能である。

【0027】UDPヘッダに関しては、IP全長フィールドと、長さがリンク層によって示されることのため、長さフィールドは冗長である。始点（送信元、ソース）がUDPチェックサムを生成しないことを選択した場合、UDPチェックサムフィールドは常に0となる。そうでない場合、無損失圧縮を保持するために、UDPチェックサムはもとのまま通信されなければならないことを発明者は認識した。

【0028】RTPヘッダに関しては、与えられたコンテキストにおいて、SSRC（同期始点：synchronization source）識別子は一定である。これは、特定のコンテキストを識別するものの一部であるからである。ほとんどのパケットの場合、シーケンス番号およびタイムスタンプのみがパケットごとに変化する。パケットがコンプレッサの上流で損失または順序誤りを生じない場合、シーケンス番号は各パケットごとに1だけインクリメントされることになる。一定期間のオーディオパケットの場合、タイムスタンプは、各パケットで運ばれるサンプル周期の数だけインクリメントされる。ビデオの場合、タイムスタンプは、各フレームの最初のパケットで変化するが、フレーム内の後続のパケットでは一定のままである。各ビデオフレームが1パケットのみを占めるが、ビデオフレームは一定レートで生成される場合も、フレームからフレームへのタイムスタンプ変化は一定である。注意すべき点であるが、これらの各場合で、シーケンス番号およびタイムスタンプフィールドの2次差分は0であるため、次のパケットヘッダは、前の非圧縮ヘッダとともにセッションコンテキストに記憶されているこれらのフィールドに対する1次差分を加えることによって、前のパケットヘッダから構成することができる。2次差分が0でない場合、変化量は通常、フィールド内の全ビット数よりもずっと小さいため、絶対値ではなく新たな1次差分を符号化し送信することによって、サイズを縮小することができる。

【0029】オーディオ有音部(talkspurt)の最初のパケットおよびビデオフレームの最後のパケットではMビットがセットされる。仮にこれを、変化ごとに完全なRTPヘッダの送信を要求するような定数フィールドとして扱えば、ヘッダ圧縮の効率が大幅に低下する。従って、以下でさらに詳細に説明するように、RTP圧縮ヘッダはMビットを明示的に運ぶことにする。

【0030】パケットがRTPミキサを通る場合（オーディオではほとんどの場合そうである）、CSRCリス

トおよびCCカウントも変化する。しかし、CSRCリストは通常、有音部以上の期間で一定であるため、変化したときのみ送信すればよい。

【0031】RTPヘッダ圧縮プロトコルの一部として使用するための、圧縮RTPヘッダの例示的なフォーマットを図14に示す。上記のように、RTPピアは圧縮RTPヘッダを送信し、ある共有情報の集まり（例えば、コンプレッサとデコンプレッサの間で一貫した状態でメモリ（図示せず）に記憶される）を維持すると仮定する。圧縮RTPヘッダは、以下のフィールドを有する。

- ・ コンテキスト更新コード（1バイト）
- ・ RTP Mビットに対するMフィールド（1ビット）
- ・ タイムクリックフィールド（7ビット）
- ・ UDPチェックサムフィールド（2バイト）
- ・ IPパケットID（2バイト）
- ・ CSRCリスト（2バイト）
- ・ RTPヘッダ拡張（2バイト）

【0032】コンテキスト更新フィールドの値は、どのような情報が図14に示すRTP圧縮ヘッダに含まれるかを示す。RTP圧縮ヘッダの最小の長さは2バイトである。RTPタイムスタンプは、タイムクリック番号（1バイト）で置換される。UDPチェックサム、IPv4パケットID、CSRCリストおよびRTPヘッダ拡張を含める必要がある場合、圧縮RTPヘッダは図14に示すよりも長くなる。しかし、ほとんどの場合、圧縮RTPヘッダは2バイトのみ（コンテキスト更新コードバイトと、M+タイムクリックバイト）となる。

【0033】各RTPピアに記憶される共有情報に関しては、IP始点および終点アドレス、UDP始点および終点アドレス、ならびにRTP SSRCフィールド（前述）の特定の組合せによって定義される、各IP/UDP/RTPパケットストリームごとに個別のセッションコンテキストがある。維持されるセッションコンテキストの数は、コンプレッサとデコンプレッサの間でネゴシエートすることが可能である。各RTPコンテキストを、GTP TID（GTPトンネルID）にマップすることが可能である（ネゴシエート可能な最大数は65536である）。各RTPヘッダ圧縮コンテキストは、固有の別個のシーケンス番号空間を有し、単一のパケット損失が1つのコンテキストしか無効にする必要がないようにしている。

【0034】各RTPヘッダ圧縮コンテキストにおける共有情報は、以下の項目を有する。

- ・ 完全なIP、UDPおよびRTPヘッダ。おそらくは、コンプレッサにより送信された、または、デコンプレッサにより再構築された最終パケットに対するCSRCリストを含む。
- ・ IPv4 IDフィールドに対する1次差分。このコンテキストに対する非圧縮IPヘッダが受信されると1

に初期化され、デルタIPV4IDフィールドが圧縮パケットで受信されるごとに更新される。

【0035】上記のように、また、図14に示すように、圧縮RTPヘッダには、RTPヘッダ（例えば、図13を参照）のRTPタイムスタンプフィールドを置き換えるタイムクリックフィールドがある。そこで、RTP受信側ピアにおいて、タイムクリックフィールド値からタイムスタンプ値およびシーケンス番号を計算（すなわち、回復）するメカニズムが必要である。このため、以下の定義をする。

s_d ：サンプル期間（単位：ms）

TS_{old} ：前のパケットのタイムスタンプ番号

TS_{new} ：このパケットのタイムスタンプ番号

WT_{old} ：前のパケットのウォールクロックの読み（ウォールクロック（壁掛け時計）とは、単に、ネットワーク基準クロックのことである）

WT_{new} ：このパケットのウォールクロックの読み

TN_{old} ：圧縮ヘッダにおける前のパケットのタイムクリック番号

TN_{new} ：圧縮ヘッダにおけるこのパケットのタイムクリック番号

SN_{old} ：前のパケットのRTPシーケンス番号

SN_{new} ：このパケットのRTPシーケンス番号

T ：サンプル期間を単位として、 n ビットを用いて表される期間（1サイクル周期）。 $T = 2^n$ サンプル（ $T = 2^n (s_d)$ ミリ秒（ms））

M ：圧縮ヘッダにおける M ビットの値

【0036】以下の式を用いて、RTP受信側ピアにおいて、タイムクリックフィールド値からタイムスタンプ値およびシーケンス番号を計算（すなわち、回復）する。

【数1】

$$TS_{new} = TS_{old} + 1 \quad (M = 0 \text{ かつ } \delta_{tick} = 1 \text{ の場合}) \quad (1)$$

$$= TS_{old} + (\delta_{cycle} \cdot T + \delta_{tick}) \quad (\text{それ以外の場合}) \quad (2)$$

$$SN_{new} = SN_{old} + \delta_{tick} \quad (3)$$

$$(\delta_{tick} \neq 1 \text{ かつ } \delta_{tick} < T/4 \text{ かつ } M \neq 1 \text{ の場合})$$

ただし

$$\delta_{tick} = (T + TN_{new} - TN_{old}) \bmod T \quad (4)$$

(ただし $\bmod T$ は T を法とする演算)

$$\delta_{utick} = (WN_{new} - WN_{old}) \bmod T \quad (5)$$

$$\delta'_{cycle} = \lfloor (WN_{new} - WN_{old}) / T \rfloor \quad (6)$$

(ただし $\lfloor \rfloor$ は 'floor' (引数以下の最大整数) を表す)

$$\delta_{cycle} = \delta'_{cycle} - 1 \quad (7)$$

($\delta_{utick} < \delta_{tick}$ かつ $\delta_{tick} - \delta_{utick} \geq T/2$ の場合)

$$= \delta'_{cycle} + 1 \quad (8)$$

($\delta_{tick} < \delta_{utick}$ かつ $\delta_{utick} - \delta_{tick} \geq T/2$ の場合)

$$= \delta'_{cycle} \quad (\text{それ以外の場合}) \quad (9)$$

【0037】有音期間中、タイムクリックフィールド値は1サンプルだけ増大する。無音期間中、タイムクリックフィールド値は、アイドル期間分（サンプル数で表される）だけ増大する。

【0038】タイムスタンプフィールドについて、解決すべき主要な問題は、7ビットタイムクリック番号がラップアラウンドした場合にどうするかである。無音期間中に、コンプレッサにより送信されるパケットがない場合、その無音期間の時間経過を（何個のクロックサイクルが経過したかに関して）検出しなければならない。例えば、この問題を解決するために、当業者に知られているウォールクロック（上記のように、例えば、 WT_{old} および WT_{new} ）が用いられる。デコンプレッサにおけるこの別個のウォールクロックが、サイクルをカウント

するために使用される。このウォールクロックは、粗い粒度（刻み、グラニュラリティ）で進行する（例えば、 $T/4$ の期間ごとに1だけ増大する。ただし、 T は、7ビットを用いて表される1サイクルの期間である）。

【0039】RTPシーケンス番号フィールドに関して、順序づけ（シーケンシング）が必要な場合、シーケンス番号を有する圧縮ヘッダが、 $T/4$ サンプルごとに、有音部の開始時に含められるべきである。必要であれば、完全なRTPヘッダを要求するために、コンテキスト修復メッセージを送信することができる。タイムクリック値が $T/4$ を超え、しかも、損失パケットがある場合には、RTP受信側がシーケンス番号情報を有するパケットを得るまで、RTPシーケンス番号を適切に更新することができない。

【0040】タイムスタンプおよびシーケンス番号の回復を行うためにこの更新アルゴリズムがどのように動作するかを以下で例示する。連続するシーケンス番号1～18を有する18個のパケットが、例えばMS205から送信されると仮定する。RTP受信側（例えば、IPエンドホスト240）では、パケット18のみが受信される（パケット7～17が失われる）とする。

【0041】第1の例として、シーケンス番号6のパケ

$$TS_{old} = 110$$

$$\delta_{tick} = (128 + 75 - 110) \bmod T = 93$$

$$\delta_{wtick} = (6(T/4) - 3(T/4)) \bmod T = 3(T/4) = 96$$

$$\delta'_{cycle} = [(6(T/4) - 3(T/4))/T] = 0$$

$$\delta_{cycle} = 0$$

$$TS_{new} = 110 + 93 = 203$$

【0042】次の第2の例では、 $TS_{old} = 38$ であり、パケット6に対するウォールクロック値は5であり、パケット18に対するウォールクロック値は9であると仮定する。パケット6および18の受信時のタイムクリック値はそれぞれ38および32である。

【数3】

$$TS_{old} = 38$$

$$\delta_{tick} = (128 + 32 - 38) \bmod T = 122$$

$$\delta_{wtick} = (9(T/4) - 5(T/4)) \bmod T = 0$$

$$\delta'_{cycle} = [(9(T/4) - 5(T/4))/T] = 1$$

$$\delta_{cycle} = 0$$

$$TS_{new} = 38 + 122 = 160$$

$$TS_{old} = 57$$

$$\delta_{tick} = (128 + 28 - 57) \bmod T = 29$$

$$\delta_{wtick} = (9(T/4) - 6(T/4)) \bmod T = 3(T/4) = 96$$

$$\delta'_{cycle} = [(9(T/4) - 6(T/4))/T] = 0$$

$$\delta_{cycle} = 1$$

$$TS_{new} = 57 + 128 + 29 = 214$$

【0044】GTPヘッダ圧縮

図15に、圧縮GTPヘッダの例示的フォーマットを示す。圧縮GTPヘッダは、バージョンフィールド（Ver s、3ビット）、ペイロードタイプフィールド（PT、1ビット）、拡張ビットフィールド（E、1ビット）、シーケンス番号フィールド（S、1ビット）、トンネル識別子（TID）存在フィールド（T、1ビット）、S NDCP存在フィールド（N、1ビット）、メッセージタイプフィールド（Msg TYPE、1バイト）、2バイトの長さ（ヘッダ長、LENGTH）フィールド、2または4バイトのトンネル識別子（TID）フィールド、2バイトのシーケンス番号フィールド、4ビットの拡張コードフィ

ットが受信されるときウォールクロック値が3（ $3(T/4)$ を意味する）であり、 $TS_{old} = 110$ であると仮定する。パケット18が受信されるとき、ウォールクロック値は6である。パケット6および18の受信時のタイムクリック値はそれぞれ110および75である。上記の式を用いると、以下の計算結果が得られる。

【数2】

【0043】次の第3の例では、 $TS_{old} = 57$ であり、パケット6に対するウォールクロック値は6であり、パケット18に対するウォールクロック値は9であると仮定する。パケット6および18の受信時のタイムクリック値はそれぞれ57および28である。

【数4】

ールド、4ビットの拡張長さフィールド、および拡張内容フィールドからなる。

【0045】TIDフィールドに関して、最上位ビットは、TIDフィールドのサイズを示すために使用される。最上位ビットの値が0の場合、TIDフィールドサイズは2バイトである。このビットの値が1の場合、TIDフィールドサイズは4バイトである。さらに、TIDフィールドは、Tビットフィールドの値がセットされている場合、すなわち、2進数1に等しい場合にのみ存在する。Eビットフィールドは、拡張ヘッダが存在するかどうかを示すために使用される。それぞれの拡張ヘッダは、4ビットの拡張ヘッダタイプ（EXT. TYPE）フィー

ルド、4ビットの拡張ヘッダ長(EXT. LENGTH)フィールド、および拡張内容(EXT. CONTENT)フィールド(そのサイズは、拡張ヘッダ長フィールドの値によって示される)からなる。拡張ヘッダ長フィールドの値は、2バイトの倍数で表される。例えば、UDPチェックサムが存在する必要がある場合、適当な拡張ヘッダタイプがこれに対して定義され、拡張ヘッダ長は1である(これは、UDPチェックサムの長さが2バイトであるため、拡張内容フィールドのサイズが2バイトであることを意味する)。

【0046】拡張ヘッダタイプフィールドの幅は4ビットであるため、16個の拡張タイプを定義することができる。拡張ヘッダ長フィールドの幅は4ビットであるため、拡張内容フィールドの最大サイズは32バイト(すなわち、2バイトの16倍)である。注意すべき点であるが、さらに多くの拡張タイプを割り当てる必要がある場合、これらのフィールドのサイズを調整(例えば、それぞれのサイズを1バイトに)することも可能である。

【0047】各GTPヘッダ圧縮コンテキスト内の(すなわち、各GTPピアに記憶された)共有情報は、以下の項目を含む(これらの項目は、最初の「完全な」GTPヘッダで受信される値に基づいて初期化される)。

- ・完全なIPおよびUDPヘッダ(この場合も、UDPチェックサムが必要な場合、受信側GTPピアは単に、受信UDPデータに基づいてそれを再計算する)。
- ・2バイトのフローラベルおよびLLCフレーム番号。
- ・TID値。

【0048】上記のように、図15に示すGTPヘッダ圧縮フォーマットは、2または4バイトの選択的なTID値をサポートする。(上記のように、TIDの97リリースバージョンは8バイトである。しかし、TID値は、さらに少ないバイト数からなるように将来再定義される可能性がある(そのため、圧縮GTPヘッダにおいては、TID値を2または4バイトのいずれかにする選択肢がある)。図15のGTPヘッダ圧縮フォーマットにおけるTIDフィールドは、TID情報を更新するために(Tビットの値を通じて)オプションとして使用される。

【0049】次に、図16に、本発明の原理に従って使用される代表的なパケットサーバの高水準ブロック図を示す。パケットサーバ605は、蓄積プログラム制御方式のプロセッサアーキテクチャであり、プロセッサ650、メモリ660(例えば、上記のGTPヘッダ圧縮などに従って通信するためのプログラム命令およびデータを記憶するためのもの)、および、バス666によって表される1つまたは複数のパケット通信ファシリティに接続するための通信インタフェース665を有する。

【0050】以上は本発明の原理の単なる例示であり、理解されるように、当業者であれば、ここに明示していませんが、本発明の技術的範囲内で、本発明の原理を実

現するさまざまな代替実施例を考えることが可能である。例えば、ここではUMTSの場合を例示したが、本発明の概念は、トンネリングプロトコルの使用を必要とする任意のワイヤレス方式(例えば、UMTSなど)あるいはアプリケーションに適用可能である。

【0051】

【発明の効果】以上述べたごとく、本発明によれば、GTP/UDP/IPヘッダの圧縮により、UMTSコアネットワークにおいて、小さいマルチメディアRTPパケットをより効率的にトランスポートすることが可能となる。

【図面の簡単な説明】

【図1】従来の圧縮されていないGTPカプセル化RTPパケットを示す図である。

【図2】本発明の原理を実現するUMTSネットワークを示す図である。

【図3】移動局で用いられる例示的なプロトコルスタックを示す図である。

【図4】移動局で用いられる例示的なプロトコルスタックを示す図である。

【図5】図2のUMTSネットワークにおける例示的なコンプレッサ/デコンプレッサ配置を示す図である。

【図6】例示的なメッセージフローを示す図である。

【図7】例示的なメッセージフローを示す図である。

【図8】例示的なメッセージフローを示す図である。

【図9】例示的なメッセージフローを示す図である。

【図10】例示的なメッセージフローを示す図である。

【図11】従来のIPヘッダフォーマットを示す図である。

【図12】従来のUDPヘッダフォーマットを示す図である。

【図13】従来のRTPヘッダフォーマットを示す図である。

【図14】圧縮RTPヘッダの例示的なフォーマットを示す図である。

【図15】圧縮GTPヘッダの例示的なフォーマットを示す図である。

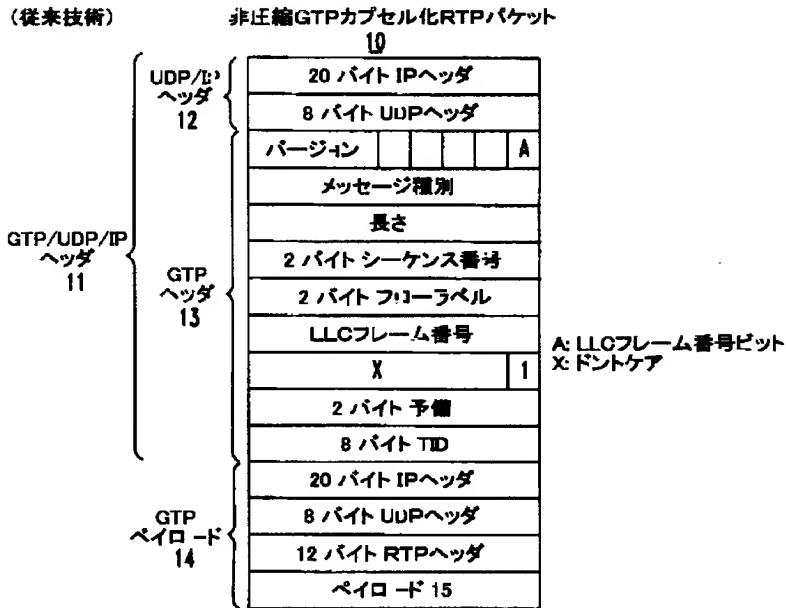
【図16】本発明の原理に従ってGTPヘッダ圧縮を実行する際に用いられるパケットサーバの例示的な高水準ブロック図である。

【符号の説明】

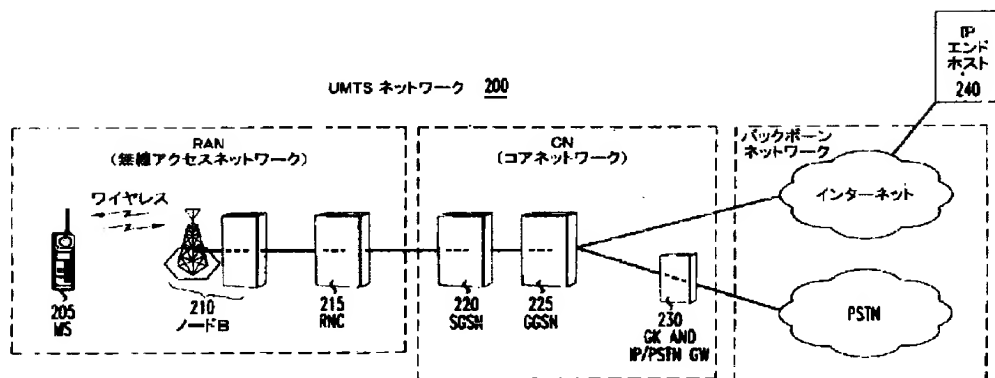
- 10 非圧縮GTPカプセル化RTPパケット
- 11 GTP/UDP/IPヘッダ
- 12 IP/UDPヘッダ
- 13 GTPヘッダ
- 14 GTPペイロード
- 15 ペイロード
- 20 バイトIPヘッダ
- 200 UMTSネットワーク
- 205 移動局(MS)

210	ノードB	240	IPエンドホスト
215	無線ネットワークコントローラ (RNC)	605	パケットサーバ
220	サービスGPRSサポートノード (SGSN)	650	プロセッサ
225	ゲートウェイGPRSサポートノード (GGSN)	660	メモリ
N)		665	通信インタフェース

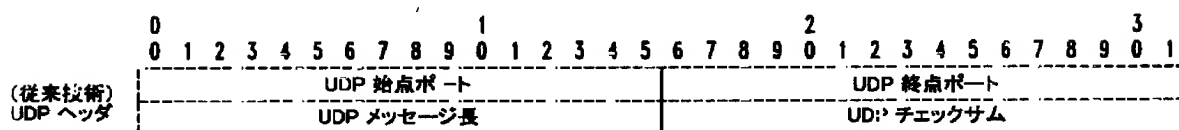
【図1】



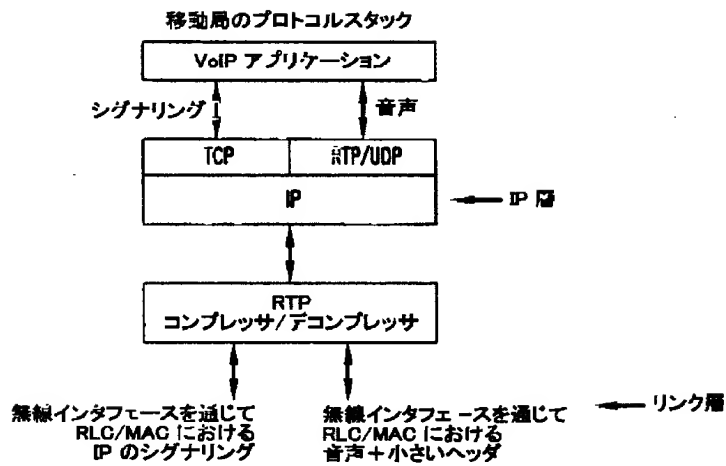
【図2】



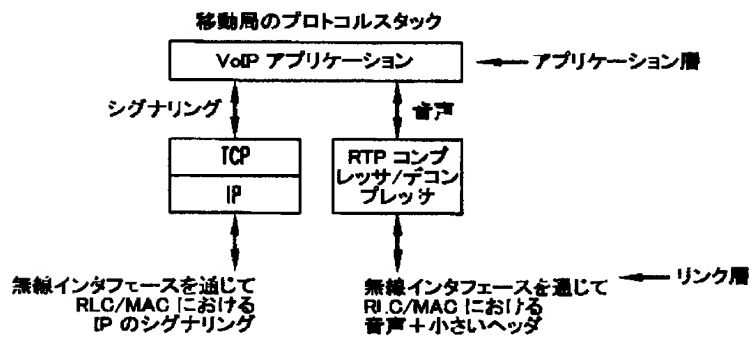
【図12】



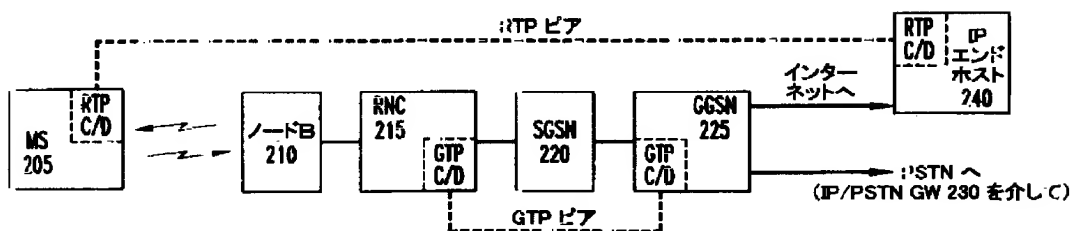
【図3】



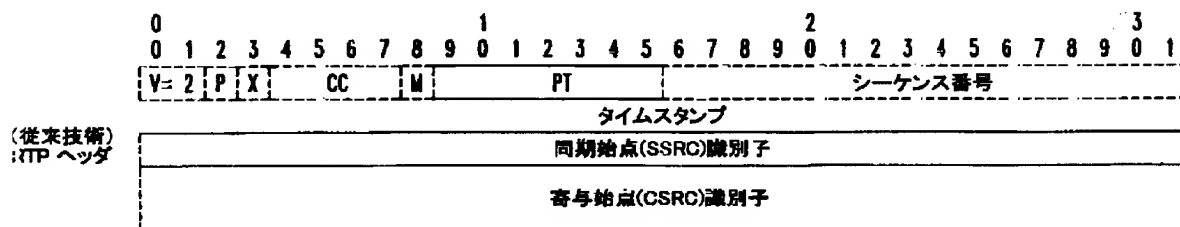
【図4】



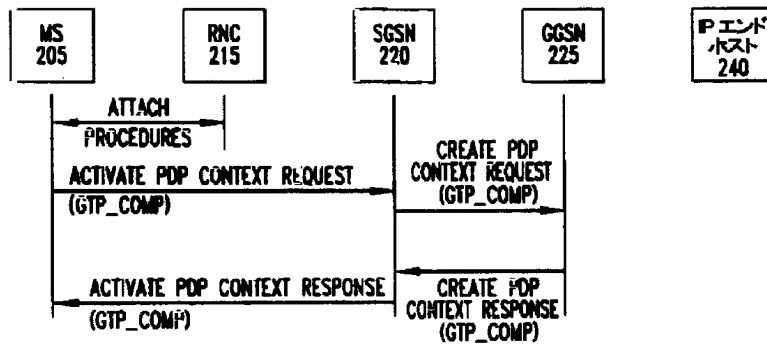
【図5】



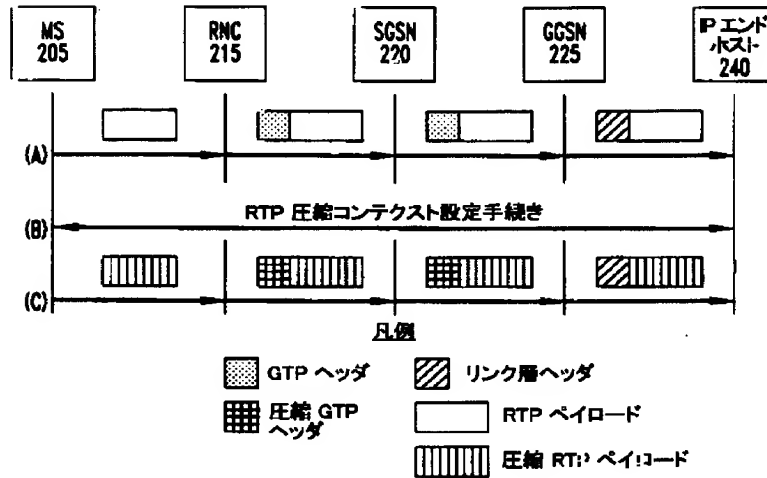
【図13】



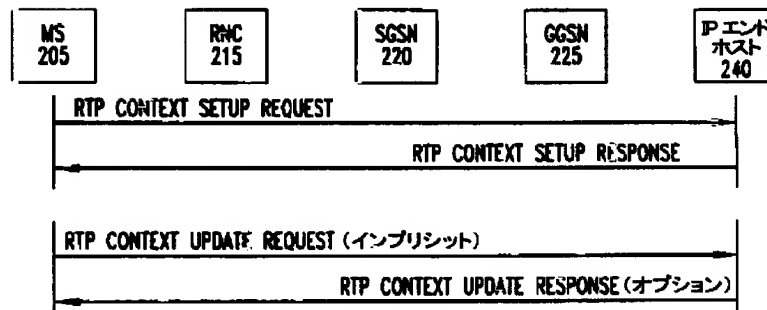
【 図 6 】



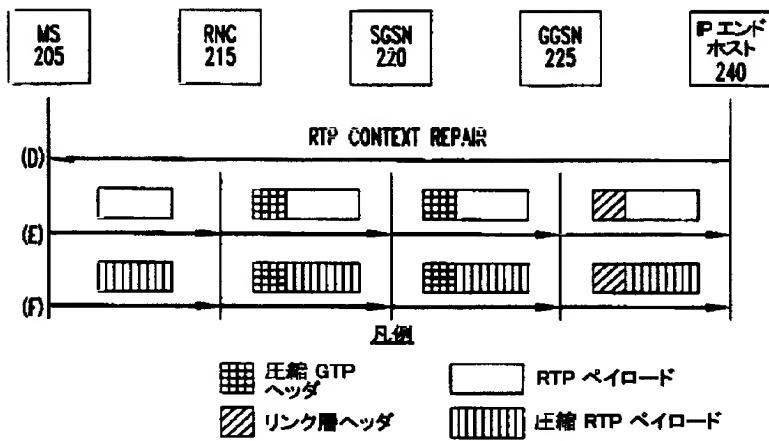
【 図 7 】



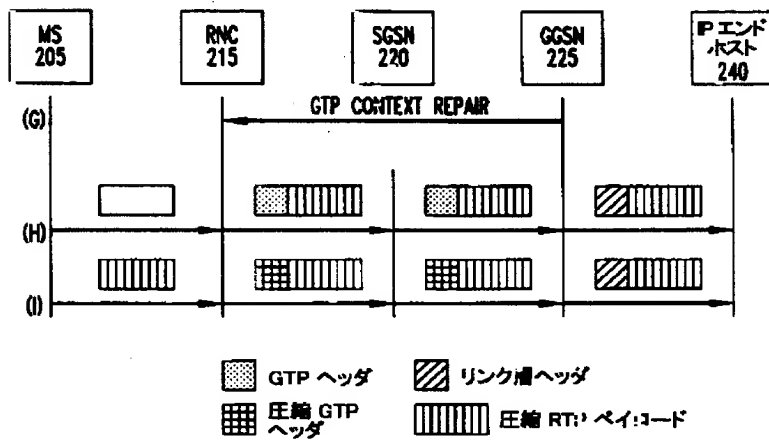
【 図 8 】



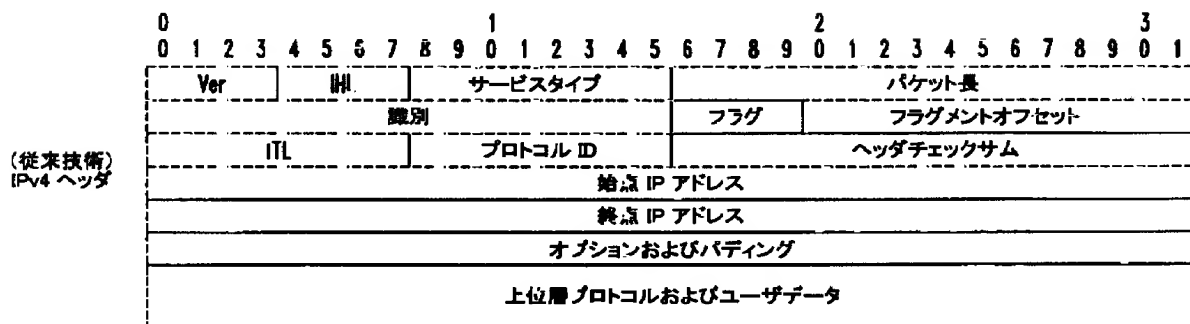
【図 9】



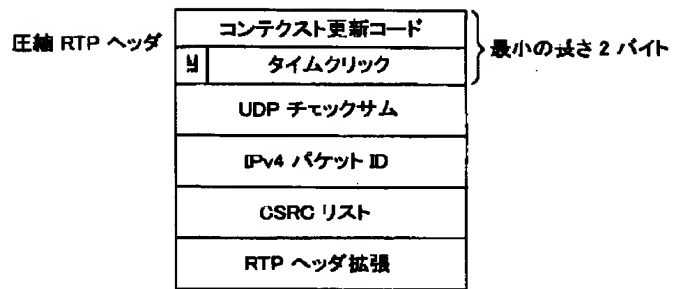
【図 10】



【図 11】

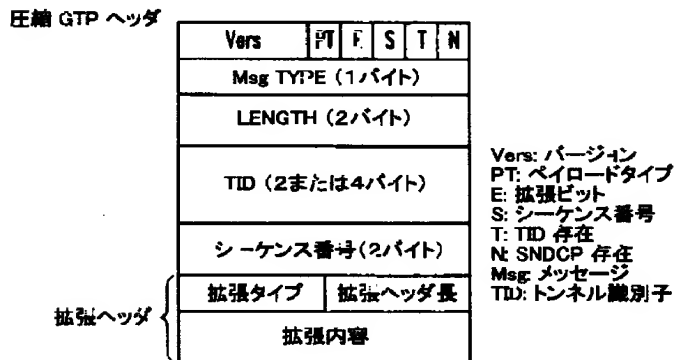


【図14】

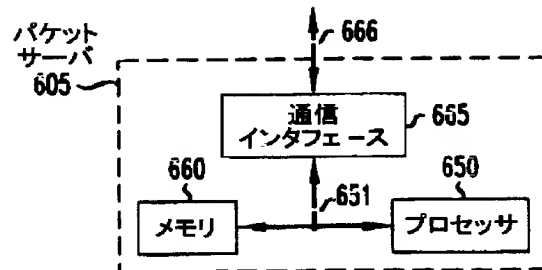


コンテキスト更新 コード
 0: M およびタイムクリックバイトのみ
 1: M + タイムクリックバイトと UDP チェックサム
 2: M + タイムクリックバイトと IPv4 パケット ID
 3: M + タイムクリックバイト, UDP チェックサムおよび IPv4 パケット ID など

【図15】



【図16】



フロントページの続き

(71)出願人 596077259

600 Mountain Avenue,
 Murray Hill, New Je
 rsey 07974-0636 U. S. A.